

---

# The "Cortex Transform" as an Image Preprocessor for Sparse Distributed Memory: An Initial Study

*Bruno Olshausen  
Andrew Watson*

1N-60  
43056  
P19

February 1990

Research Institute for Advanced Computer Science  
NASA Ames Research Center

RIACS Technical Report 90.4

NASA Cooperative Agreement Number NCC2-408 and NCC 2-387

(NASA-CR-187436) THE CORTEX TRANSFORM AS AN  
IMAGE PREPROCESSOR FOR SPARSE DISTRIBUTED  
MEMORY: AN INITIAL STUDY (Research Inst.  
for Advanced Computer Science) 19 pCSCL 09B

N91-32798

Unclass  
G3/60 0043056

# RIACS

**Research Institute for Advanced Computer Science**  
An Institute of the Universities Space Research Association

---



# **The "Cortex Transform" as an Image Preprocessor for Sparse Distributed Memory: An Initial Study**

*Bruno Olshausen \**  
*Andrew Watson †*

**Research Institute for Advanced Computer Science  
MS 230-5, NASA Ames Research Center  
Moffett Field, CA 94035**

**RIACS Technical Report 90.4**

**February 1990**

**Abstract.** We describe an experiment designed to evaluate the use of the "Cortex Transform" (Watson, 1987) as an image preprocessor for Sparse Distributed Memory. In the experiment, a set of images were injected with Gaussian Noise, preprocessed with the Cortex Transform, and then encoded into bit patterns. The various spatial frequency bands of the Cortex Transform were encoded separately so that they could be evaluated based on their ability to properly cluster patterns belonging to the same class. The results of this study indicate that by simply encoding the low-pass band of the Cortex Transform, a very suitable input representation for the SDM can be achieved.

---

**This research was supported in part by cooperative agreements NCC2-408 and NCC2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA), and in part by NASA RTOP 505-67.**

**\* Olshausen, a graduate student at Caltech, was a Research Associate at RIACS during the course of his study.**

**† Watson is a NASA Scientist at the Ames Research Center.**



# The "Cortex Transform" as an Image Preprocessor for Sparse Distributed Memory: An Initial Study

*Bruno Olshausen*

*Research Institute for Advanced Computer Science*

*and*

*Andrew Watson*

*NASA Ames Research Center*

**Abstract.** We describe an experiment designed to evaluate the use of the "Cortex Transform" (Watson, 1987) as an image preprocessor for Sparse Distributed Memory. In the experiment, a set of images were injected with Gaussian noise, preprocessed with the Cortex Transform, and then encoded into bit patterns. The various spatial frequency bands of the Cortex Transform were encoded separately so that they could be evaluated based on their ability to properly cluster patterns belonging to the same class. The results of this study indicate that by simply encoding the low-pass band of the Cortex Transform, a very suitable input representation for the SDM can be achieved.

Sparse Distributed Memory (SDM), an associative memory described by Kanerva (1988), is well-suited to perform high-level object recognition tasks because of its ability to quickly classify patterns on the basis of incomplete or corrupted information. This ability would be especially useful for visual recognition tasks, where typically an object in a scene must be quickly identified despite the presence of noise and distortions in the imaging process, or variations in the shape of the object. However, before SDM can be applied to visual object recognition problems, it is necessary to determine how raw images should be preprocessed and encoded in order to form a suitable input for the SDM.

To determine how raw images should be processed, it is first necessary to consider what types of variations in the image may interfere with the proper classification of an object. In this case, since we are interested in applying SDM to the problem of recognizing 2D shapes, we need to be concerned with such image variations as pixel noise, changes in contrast, line thickness, or even slight variations in the shape's structure (e.g., hand-drawn characters). At this stage, however, we concern ourselves with the case of pixel noise only (i.e., an independent and identically distributed Gaussian process added to each image pixel value).

Because the SDM uses the Hamming distance between two bit-patterns as a measure of their "closeness," our goal in preprocessing and encoding the image is to develop a bit-string representation of the image such that two shapes belonging to the same class give rise to bit-strings that are close in Hamming distance. Conversely, shapes belonging to different classes should give rise to bit-strings that are well-separated in Hamming distance. In this paper, we examine how well the Cortex Transform (Watson,

1987), serving as the preprocessor, accomplishes this goal for images that have been perturbed with pixel noise only.

## The Cortex Transform

The Cortex Transform is described in detail by Watson (1987, 1988). Here we discuss only the important features that were used in the experiment.

The Cortex Transform subdivides the content of an image into different spatial-frequency bands by filtering the image with a set of oriented, bandpass filters as shown in Figure 1. This process converts a single image into multiple images, each of which contains a unique subset of the spatial frequencies present in the original image. When properly sub-sampled, these images can provide a very compact representation of the original image because their pixels have very little correlation with one another. This property is not only highly desirable for image compression, but would also be useful in preprocessing images for SDM. This is because in the encoding process, we wish to maximize the information content of each pixel being encoded. If the pixels are uncorrelated to one another, then each pixel is "saying" the most it can about the content of the image.

In general, the Cortex Transform produces enough output images to give a complete representation of all the different spatial frequency bands in an image, such as depicted in Figure 1. For our purposes though, we chose to use only a portion of the bands. These are two different band-pass filters subdivided into four orientations each, and two different low-pass filters, as shown in Figure 2. Note that each set of bandpass filters results in a set of four images - one image for each orientation - while each of the low-pass filters results in only one image. All the filtered images were sub-sampled so as to reduce the number of pixels with only a negligible loss of information (see Watson, 1988).

While the output images of the Cortex Transform contain both the magnitude and phase of spatial frequency components, we considered only the magnitude to be important. Our reason for this is that we were not interested in the exact location of spatial frequency components (which the phase would provide), but more importantly, the extent to which they are present in the image and their approximate location.

## Encoding the Cortex Transform

The magnitudes of the filtered images were encoded into bit-strings by first quantizing the pixel values of an image into five levels each and then using a 4-bit thermometer-type code, as shown in Table 1, to encode each pixel. The quantization levels for an image were determined by finding the minimum and maximum pixel values in the image, and then setting the quantization thresholds at uniform intervals over the range from minimum to maximum. A set of four images (corresponding to four orientations) was similarly quantized, except that the minimum and maximum pixel

pixel value	bit-string
0	0000
1	0001
2	0011
3	0111
4	1111

Table 1: Encoding quantized pixel values into bit-strings.

values were computed over the set of all four images, and the ensemble was quantized as a whole.

A bit-string representation of the image was then formed by simply concatenating the 4-bit codes for each pixel into one long bit-string. Thus, an  $N \times N$  low-pass filtered image would result in a bit-string of length  $N \times N \times 4$ , and four (oriented)  $N \times N$  band-pass filtered images would result in a single bit-string of length  $N \times N \times 16$ .

## Experiments

The set of shapes used in this experiment were the 26 capital letters of the alphabet, extracted from the Courier 24-point font on a SUN 3/60 workstation, as shown in Figure 3. Each image contained only two pixel values: 0 for off (white) and 255 for on (black). These images were then injected with noise by adding an i.i.d. Gaussian process to the pixel values. (Note: since the output of the noise injection process was a floating point image, no special action needed to be taken for pixel values lying outside the interval  $[0, 255]$ .) This was done using standard deviations ( $\sigma$ ) of 40, 80, and 120. For each standard deviation, 20 samples of noise were generated. Thus, each letter had a total of 60 different noisy instances ( $3 \text{ values of } \sigma \times 20 \text{ instances}/\sigma$ ). These images, in addition to the 26 original (non-noisy) images, were then processed with the Cortex Transform under the four filter arrangements shown in Figure 2. The filtered images were then sub-sampled and encoded 4-bits/pixel as described above. Since the original images were padded in a  $32 \times 32$  square, only the central regions of each sub-sampled, filtered image needed to be encoded, as shown in Figure 4.

This entire process is illustrated in Figure 5.

Once the images were encoded, the resulting bit-patterns were compared by Hamming distance to determine how well patterns of the same class were clustered together. This was done as follows:

Assuming we have chosen a particular standard deviation of noise,  $\sigma$ , and a particular filtering strategy (one of the four shown in Fig. 2), then we denote the  $j^{\text{th}}$  noisy instance of pattern  $i$  as,

$$P_{ij} \quad [i=1\dots 26, j=1\dots 20],$$

and we denote the original (non-noisy) instance of pattern  $i$  as,

$$P_i^* \quad [i=1\dots 26],$$

( $P$  is a bit-pattern obtained by encoding a preprocessed image).

Then, we denote the Hamming distance between an original (non-noisy) instance of pattern class  $k$  and the  $j^{\text{th}}$  noisy instance of pattern class  $i$  as,

$$d_{kij} = |P_k^* - P_{ij}|_{L1}$$

where  $| \cdot |_{L1}$  denotes the L1 norm (Hamming distance for bit-strings). These

distances were computed for all  $k, i, j$ . The distances were then accumulated into two different histograms for each  $k$ : one histogram for comparisons between  $P_k^*$  and instances within class  $k$  (which we denote  $H_{in\_k}$ ), and another for comparisons between  $P_k^*$  and instances from classes other than  $k$  (which we denote  $H_{not\_in\_k}$ ). Formally, this can be expressed as,

$$H_{in\_k}(d) = \sum d_{kij}, \text{ (sum over } i=k, j=1\dots 20),$$

and

$$H_{not\_in\_k}(d) = \sum d_{kij}, \text{ (sum over all } i \neq k, j=1\dots 20).$$

Then, by integrating each histogram up to some distance,  $D$ , we obtain a measure of the signal-to-noise ratio that would result when reading the SDM with Hamming radius  $D$ . That is,

$$S_{in\_k}(D) = \sum H_{in\_k}(d), \text{ (sum over } d=0\dots D),$$

and,

$$S_{not\_in\_k}(D) = \sum H_{not\_in\_k}(d), \text{ (sum over } d=0\dots D).$$

If  $S_{in\_k}(D)$  is less than  $S_{not\_in\_k}(D)$ , this would indicate that there are more instances of class  $k$  than of any other class within  $D$ . In terms of reading from the SDM, then, this would mean that there is at least a possibility of recovering the correct data when reading the memory with  $P_k^*$  as the address and using Hamming radius  $D$ . Otherwise, the data corresponding to  $P_k^*$  would most certainly be overwhelmed by data from other classes. Thus, we use the function

$$p_k(D) = \begin{cases} 1, & \text{if } S_{in\_k}(D) < S_{not\_in\_k}(D) \\ 0, & \text{otherwise,} \end{cases}$$

to denote whether class  $k$  "passes" (1) or "fails" (0) at Hamming distance  $D$ . Then, in order to get a global measure of performance at Hamming distance  $D$ , we average  $p_k(D)$  over all classes:

$$Perf(D) = 1/26 \sum p_k(D), \text{ (sum over } k=0\dots 26).$$

This function provides us with a reasonable measure by which to judge the representation formed by each of the filter processes.



## Results

The function  $Perf(D)$  is plotted in Figures 6-11 for the different standard deviations of noise ( $\sigma=40, 80, 120$ ) and filtering strategies tested. Figures 6-8 plot the performance of the low-pass filters, and Figures 9-11 plot the performance of the oriented, band-pass filters along with the performance of the raw image (i.e., no preprocessing)

For nearly all cases, the lowest-pass filter (i.e., the filter with the lowest cutoff frequency) provided the best performance, yielding 96% "passes" at  $D=5$  bits (Fig. 6). This filter has a cutoff of 0-0.125 cycles/pixel and compresses the original  $17 \times 15$  pixels to  $5 \times 4$ , yielding a 12-fold decrease in the number of pixels (and hence the number of bits in the SDM input bit-string).

Note that with a high standard-deviation of noise ( $\sigma=120$ ), the lowest-pass band no longer provides the best performance (Fig. 8). The best performance is provided instead by the next lowest band (0-0.25 cycles/pixel). Also at this noise level, the performance of the highest-pass band filter (0.25-0.5 cycles/pixel) is worse than that of the raw image with no preprocessing (Fig. 11). However, in all other cases, preprocessing yields an improvement in the image representation.

## Conclusion

Our goal in this experiment was to evaluate the Cortex Transform as a preprocessor for Sparse Distributed Memory. The results indicate that for 2-D shapes for which the image has been perturbed with pixel noise only, a dramatic improvement in the image representation may be obtained by encoding the low spatial-frequency bands of the Cortex Transform.

It should be noted that this experiment was intended as an initial study to evaluate the use of multi-resolution or oriented filters with SDM. There are many further extensions to this work. One would be to compute the performance of the various filters for other image variations, such as line-thickness, contrast, or small structural variations. Another possibility would be to examine other encoding strategies, such as the use of real numbers instead of bits. In this case, it may also be useful to investigate the effect of using an L2 distance metric instead of L1.

An important advantage of conducting tests such as described here is that it is possible to evaluate a representation by itself without getting involved with the implementation details of SDM. This allows one to focus on the special problems of the application domain - such as in this case, the variations that can take place in an image - before employing more powerful, higher-level machinery.

## Acknowledgments

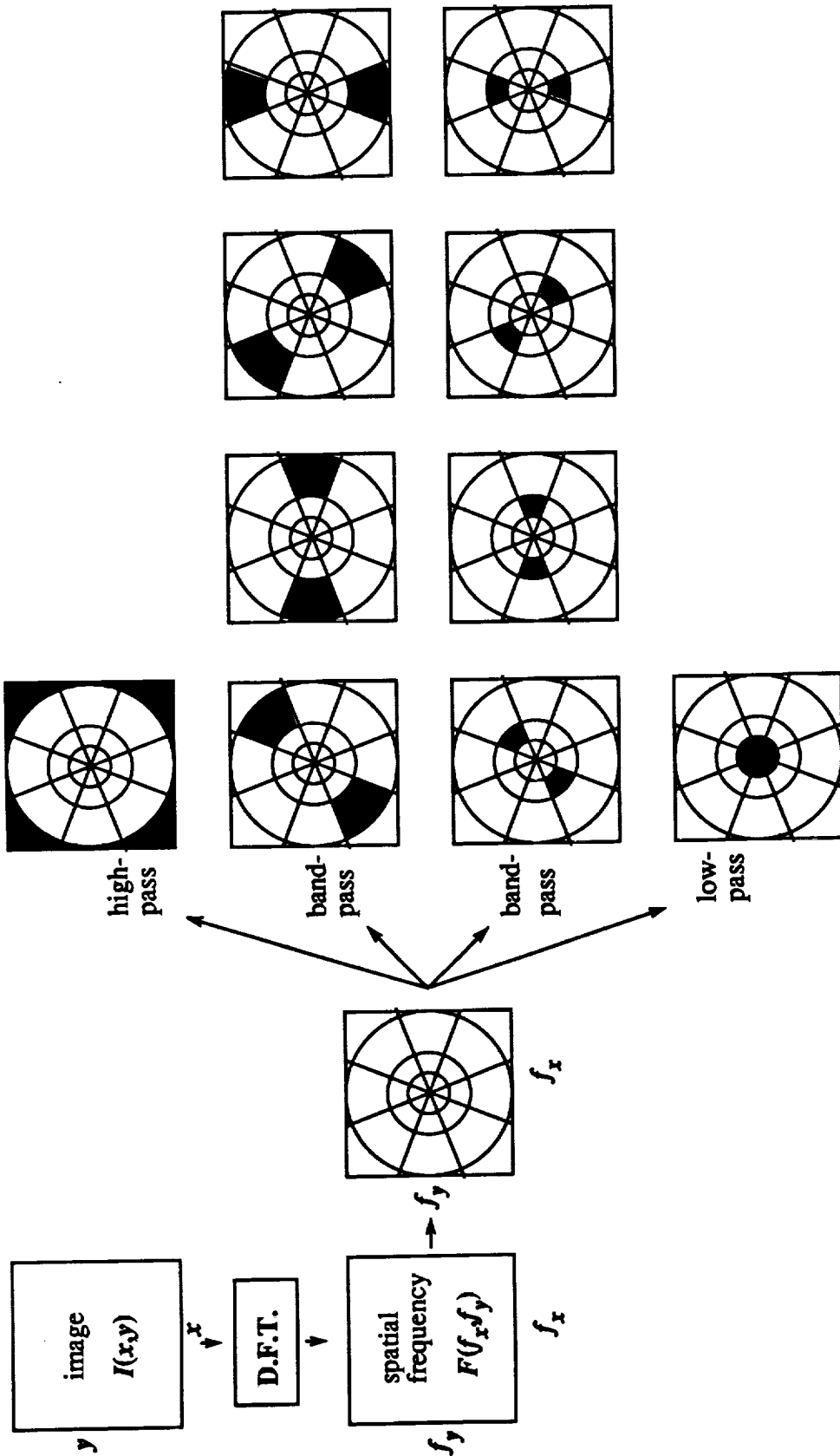
David Rogers was extremely helpful in suggesting various performance metrics for this experiment.

## References

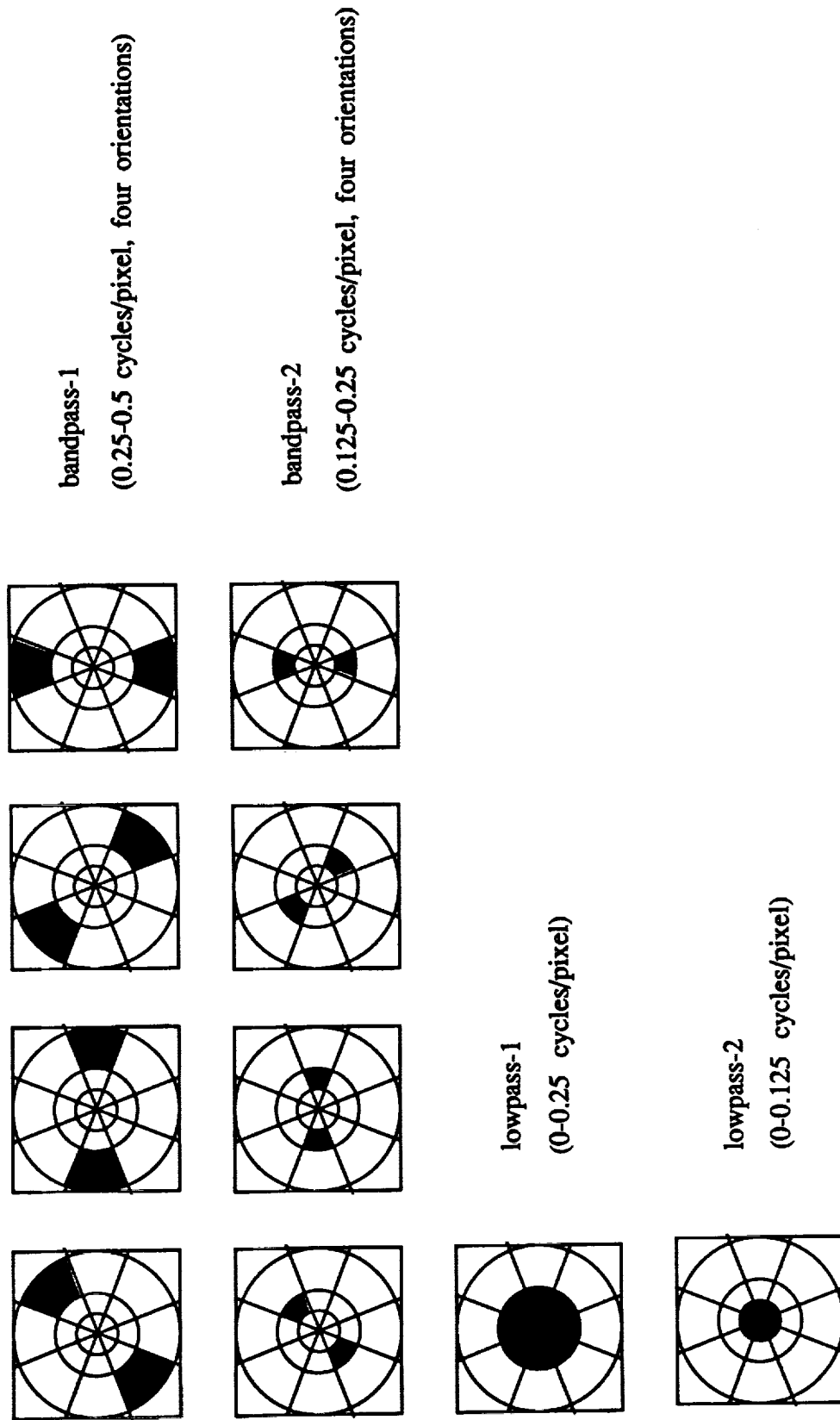
Kanerva, P., *Sparse Distributed Memory*, MIT Press, 1988.

Watson, A.W., The Cortex Transform, *Computer Vision, Graphics, and Image Processing*, 1987.

Watson, A.W., Efficiency of a model human image code, *Journal of the Optical Society of America, A*, Vol. 4, No. 12, Dec. 1987, pp. 2401-2417.



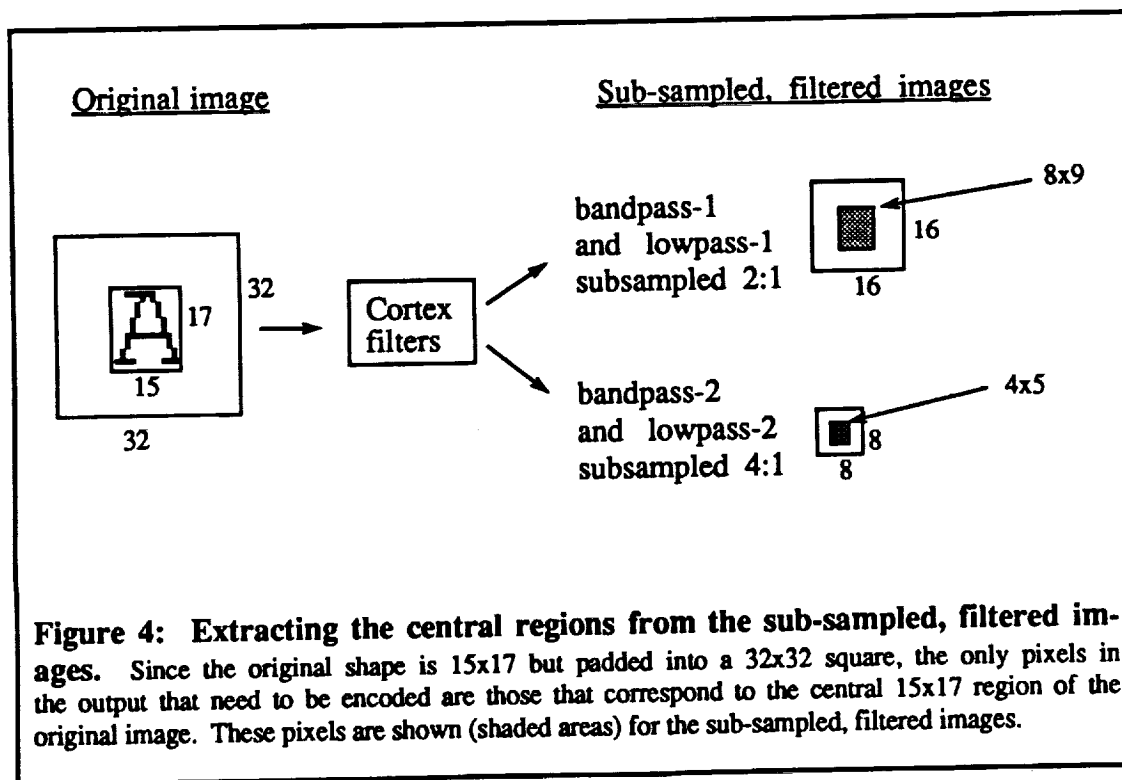
**Figure 1: The Cortex Transform.** The content of an image is subdivided into different spatial frequency bands by slicing-up the spatial frequency plane according to both frequency and orientation. The D.F.T. (Discrete Fourier Transform) of an image  $I(x,y)$  is shown here as  $F(x,y)$ .  $F(x,y)$  is then subdivided as shown: black regions are where the spatial-frequency coefficients are left unaltered; white regions are where the spatial frequency coefficients are set to zero. Thus, the inverse D.F.T. of each of these spatial-frequency "images" will result in a band-pass (or low-pass) filtered version of the original image (not shown). Thus, a single image is converted into multiple images, each of which carries a unique portion of the frequency/orientation content of the original.



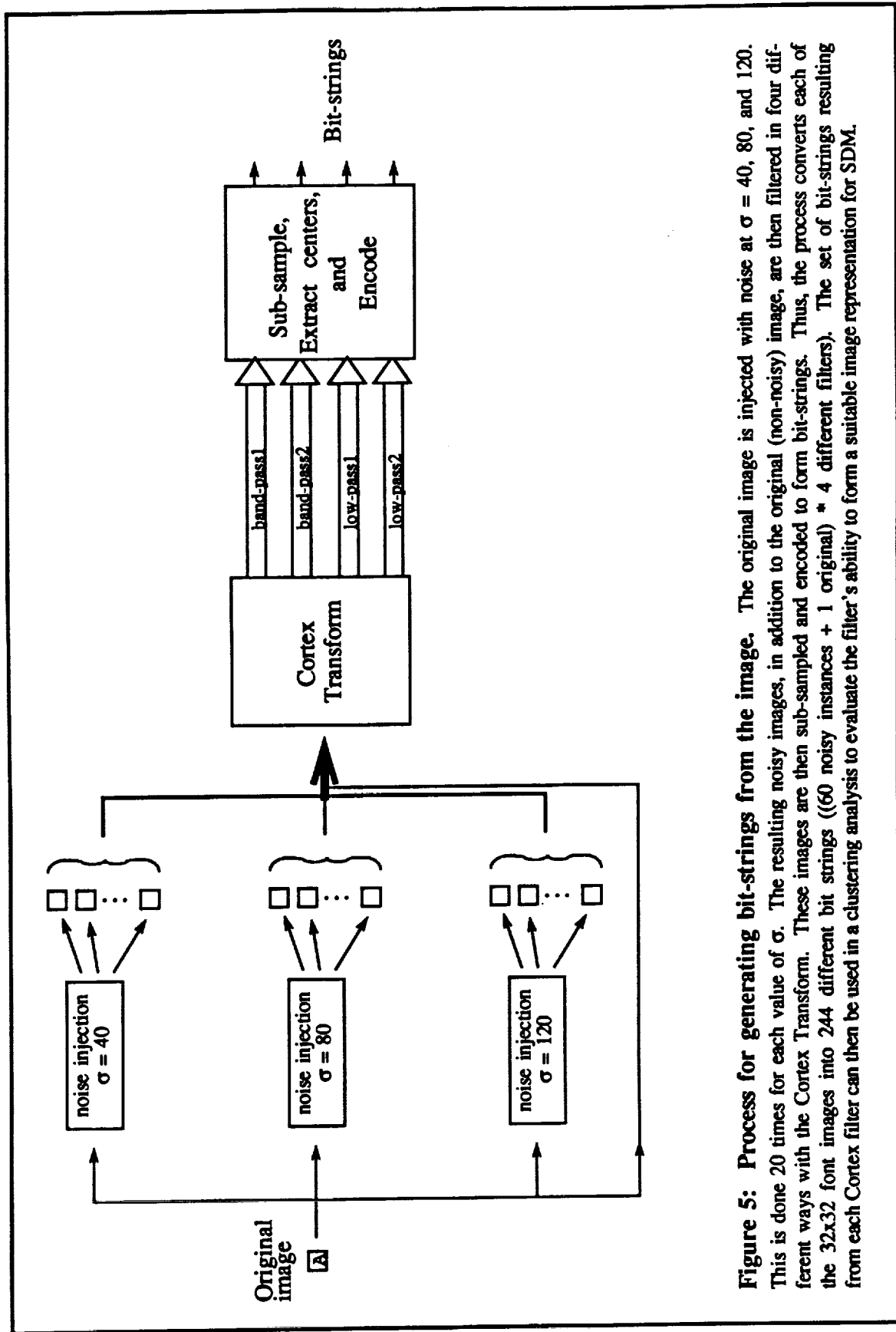
**Figure 2: The Cortex filters used in the experiment.** There are two sets of oriented, band-pass filters an octave apart, and two different low-pass filters an octave apart. The original image contains 32x32 pixels; thus, each spatial-frequency "image" also contains 32x32 pixels (in this case, spatial frequency coefficients). Note that each set of four band-pass filters will result in a set of four filtered images - one for each orientation - while each of the low-pass filters will result in only one image.

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S	T	U
V	W	X	Y	Z		

**Figure 3:** The set of shapes used in the experiment. These character shapes were extracted from the Courier 24-point font on a SUN 3/60 workstation. Each font is 15x17, but padded by zeros into a 32x32 image so as to avoid "wrap-around" effects with the Cortex Transform.



**Figure 4:** Extracting the central regions from the sub-sampled, filtered images. Since the original shape is 15x17 but padded into a 32x32 square, the only pixels in the output that need to be encoded are those that correspond to the central 15x17 region of the original image. These pixels are shown (shaded areas) for the sub-sampled, filtered images.



**Figure 5: Process for generating bit-strings from the image.** The original image is injected with noise at  $\sigma = 40, 80,$  and  $120$ . This is done 20 times for each value of  $\sigma$ . The resulting noisy images, in addition to the original (non-noisy) image, are then filtered in four different ways with the Cortex Transform. These images are then sub-sampled and encoded to form bit-strings. Thus, the process converts each of the  $32 \times 32$  font images into 244 different bit strings ((60 noisy instances + 1 original) \* 4 different filters). The set of bit-strings resulting from each Cortex filter can then be used in a clustering analysis to evaluate the filter's ability to form a suitable image representation for SDM.

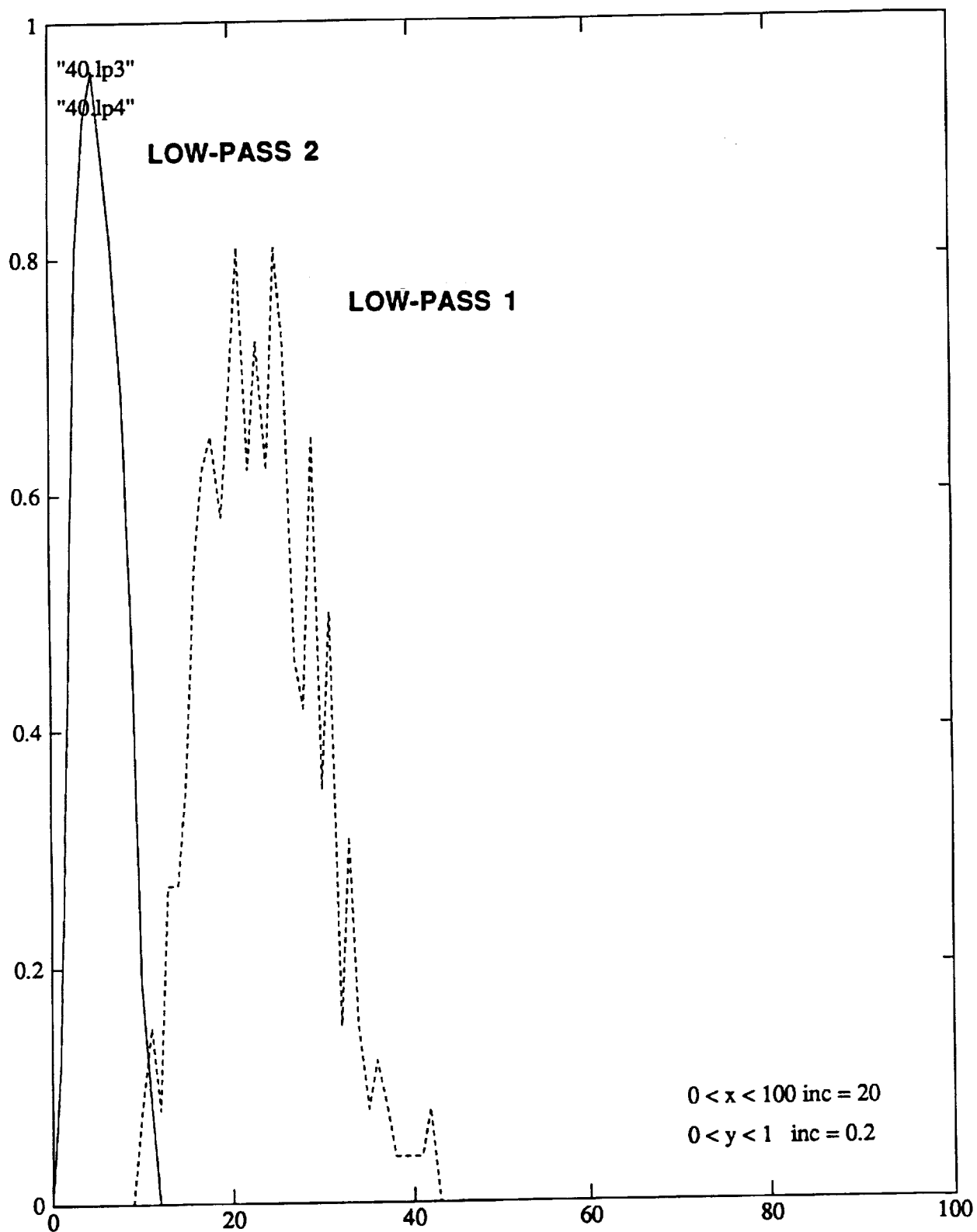


Figure 6: Performance of the low-pass filters at  $\sigma = 40$ .

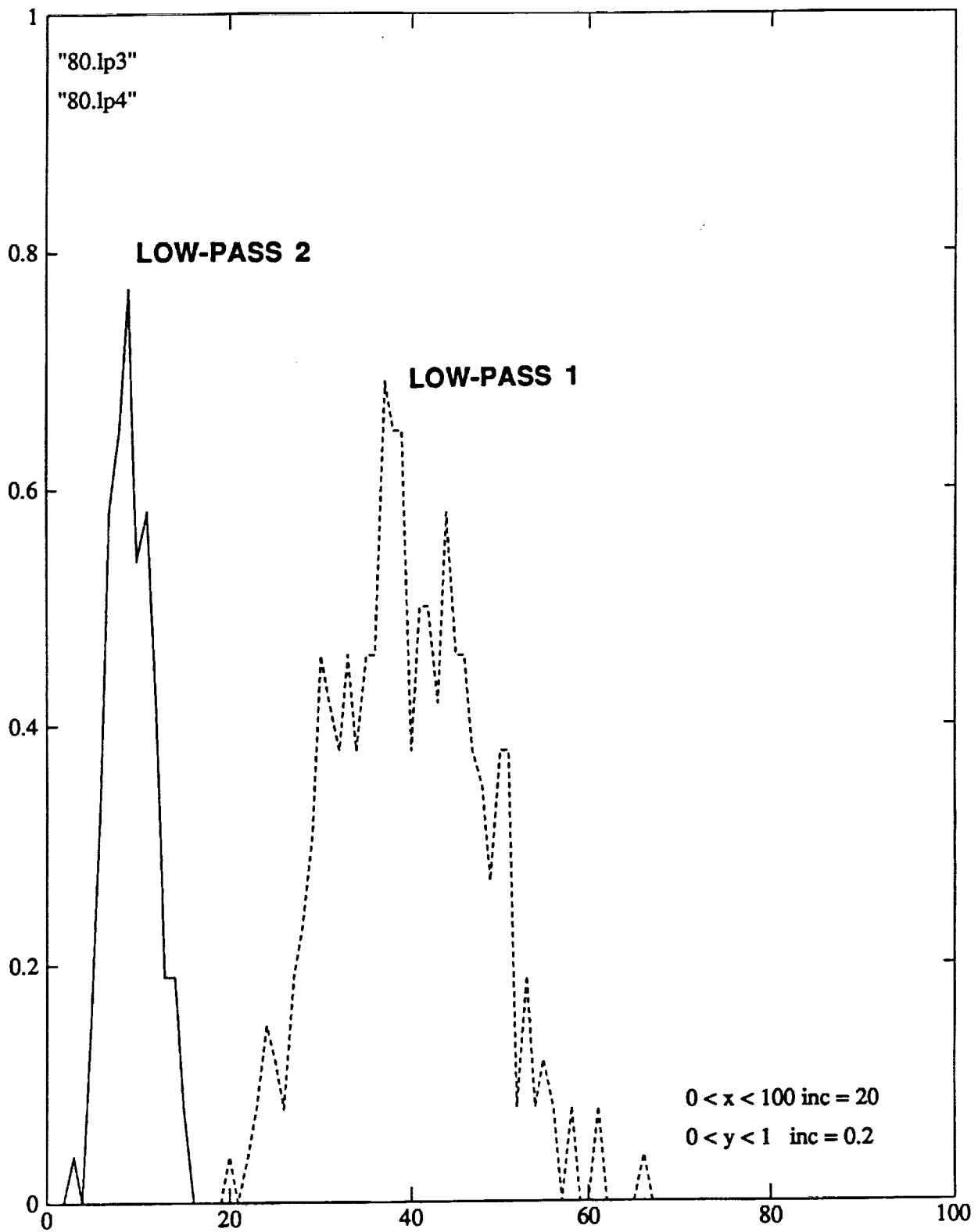


Figure 7: Performance of the low-pass filters at  $\sigma = 80$ .



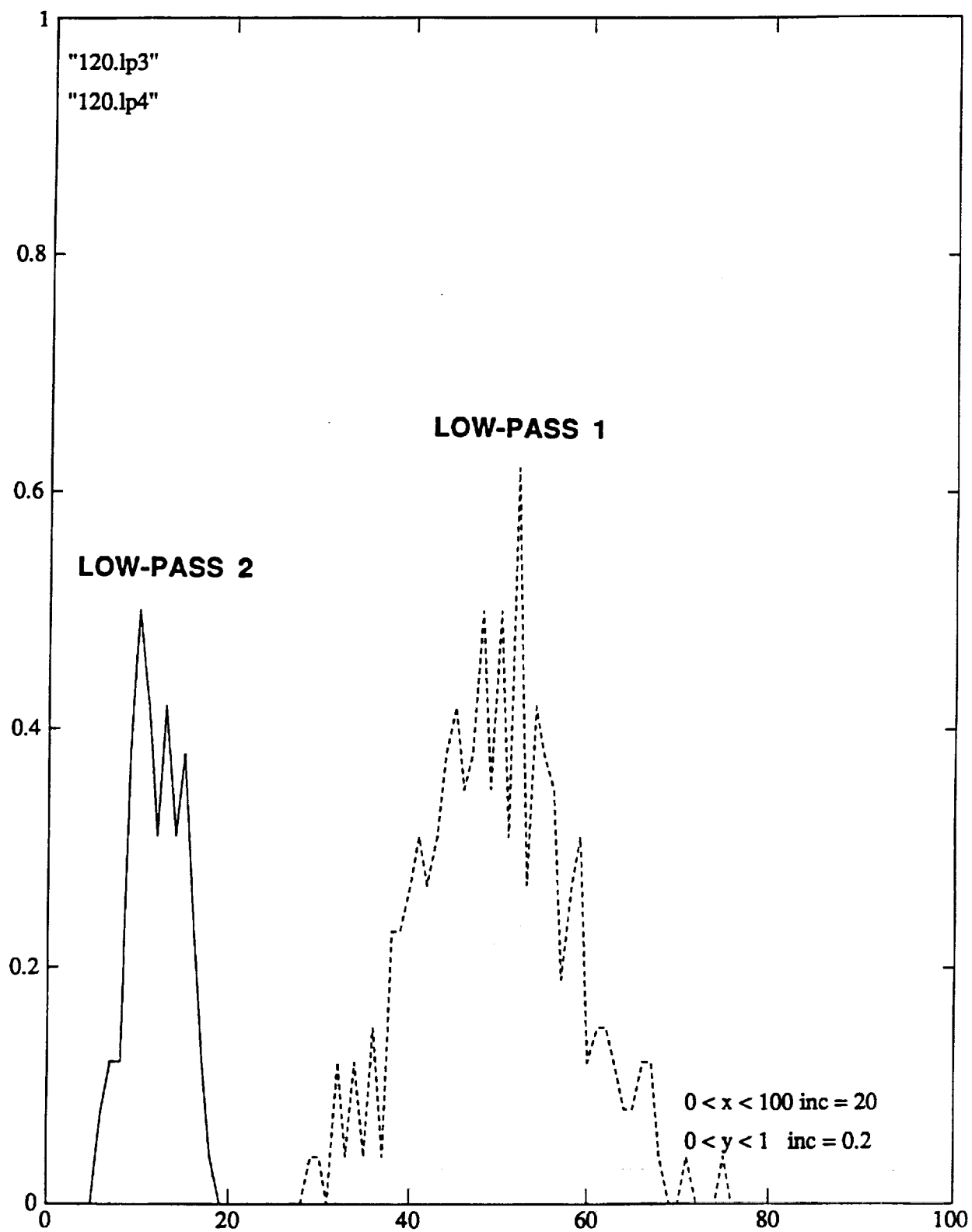


Figure 8: Performance of the low-pass filters at  $\sigma = 120$ .

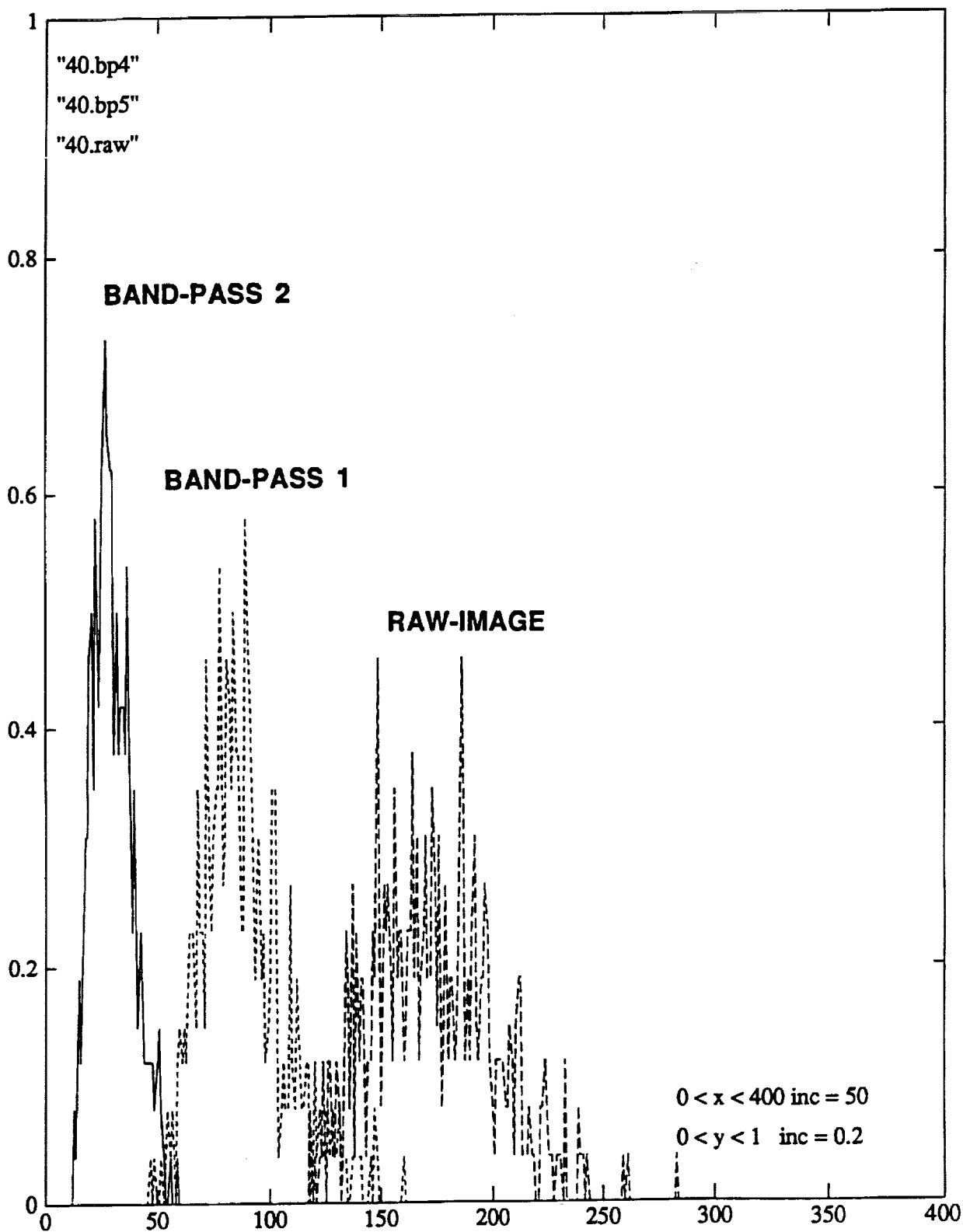
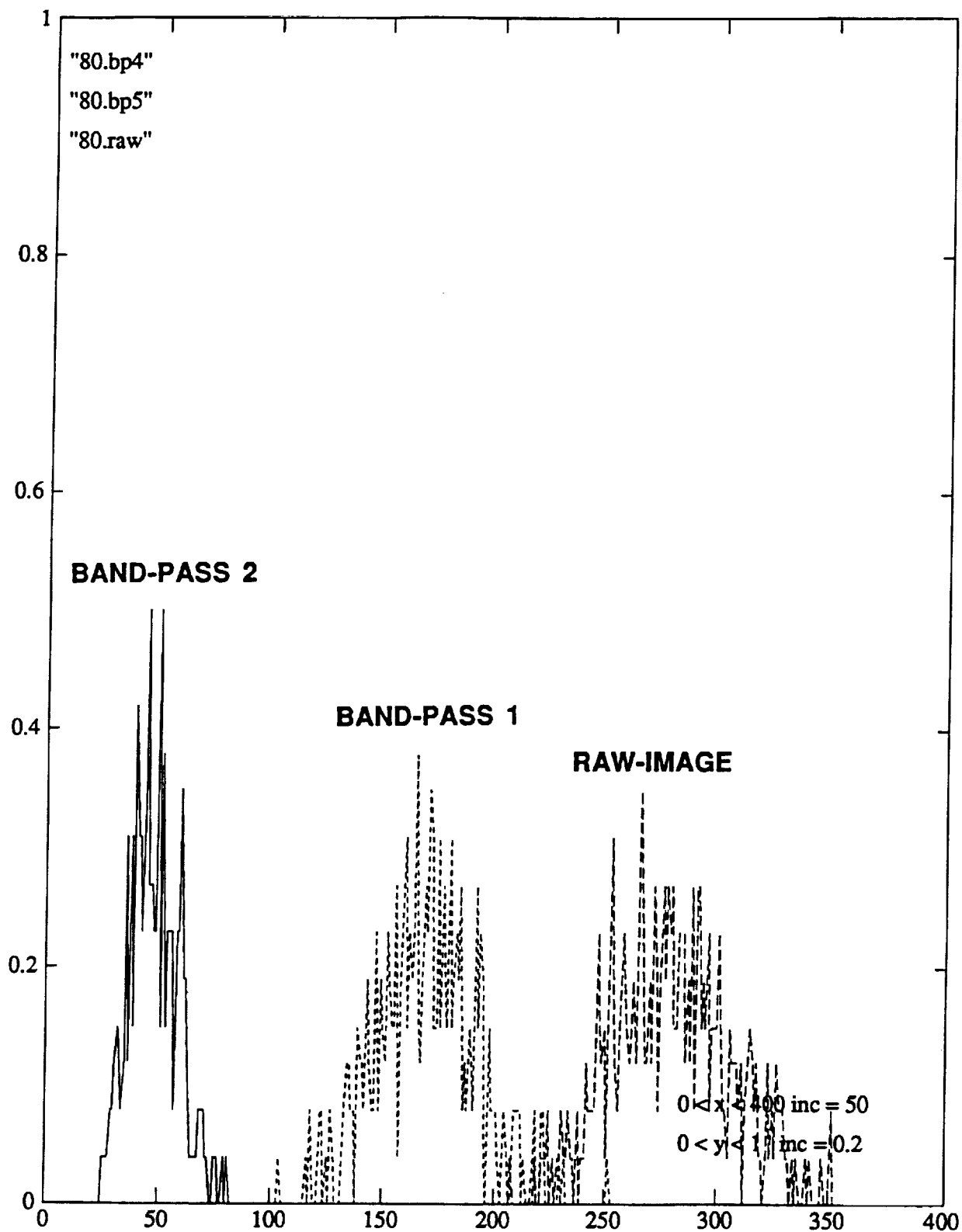


Figure 9: Performance of the band-pass filters and raw-image at  $\sigma = 40$ .



**Figure 10: Performance of the band-pass filters and raw-image at  $\sigma = 80$ .**

